

chsos20122908514

SISTEMAS OPERATIVOS

NOVIEMBRE 2, 2012 de BRIGETHCC

SISTEMAS OPERATIVOS EMBEBIDOS

SISTEMAS OPERATIVOS EMBEBIDOS, CE Y TIEMPO REAL

CARACTERÍSTICAS GENERALES

SISTEMAS OPERATIVOS EMBEBIDOS

Los sistemas embebidos y los de tiempo real son tecnologías inmersas en el diario vivir y generalmente no se identifican fáciles y tampoco se definen los sistemas operativos que hacen posible su manipulación y que a la vez administran los recursos de dichos dispositivos para maximizar el desempeño.

Un sistema operativo embebido es aquel que está integrado en los circuitos de los dispositivos electrónicos, entre estos dispositivos se encuentra electrodomésticos, teléfonos móviles, radios, televisores, automóviles, lectores de códigos de barras, equipos médicos, asistentes digitales personales (PDA), etc.

Estos sistemas suelen tener algunas características de los sistemas de tiempo real pero también tienen limitaciones de tamaño, memoria y consumo de

electricidad que los hace especiales y no suelen ser visibles.

Algunas características son:

Fiabilidad y seguridad: Un fallo en un sistema de control puede hacer que el sistema controlado se comporte de forma peligrosa o antieconómica. Es importante asegurar que si el sistema de control falla lo haga de forma que el sistema controlado quede en un estado seguro, hay que tener en cuenta los posibles fallos o excepciones en el diseño

Eficiencia: Gran parte de los sistemas de control deben responder con gran rapidez a los cambios en el sistema controlado

Interacción con dispositivos físicos: Los sistemas empotrados interaccionan con su entorno mediante diversos tipos de dispositivos que normalmente no son convencionales (teclados, impresoras): convertidores A/D y D/A, pwm, entradas y salidas digitales paralelo y serie, (interfases con sensores, actuadores, periféricos especiales). Los componentes del software que controlan el funcionamiento de estos dispositivos son, en general, dependientes del sistema concreto.

Robustez: Embarcados en sistemas con movimiento o que pueden ser transportados, sujetos a vibraciones e incluso impactos (coches, robots, instrumentación portátil). No siempre trabajan en condiciones óptimas de temperatura, humedad, y limpieza.

Entre las plataformas de los sistemas operativos embebidos tenemos:

QNX es un sistema operativo embebido desarrollado por QNX Software Systems Ltd, para aplicaciones de dispositivos electrónicos, telecomunicaciones, sistemas automotrices, etc.; que necesitan una gran confiabilidad, desempeño, funcionalidades específicas, y una escalabilidad masiva.

Linux embebido es un sistema Linux típico del cual se han removido programas de utilidad, herramientas, y otros servicios del sistema que no son necesarios en un ambiente embebido. Linux para los dispositivos embebidos comenzó con el soporte del kernel y el compilador para los microprocesadores más populares de 32 bits: x86, ARM, PowerPC, MIPS y SH. Y luego continuó con la aparición de diferentes distribuciones de Linux con soporte para características específicas de los sistemas embebidos.

Windows XP para sistemas embebidos. Aunque Microsoft ha mantenido a los fabricantes de sistemas operativos alternativos a Windows contra la pared en lo que se refiere a equipos de sobremesa, su posición en la batalla por la supremacía en el terreno de los sistemas operativos para dispositivos que no fueran PC era vulnerable. De ahí que Microsoft da un paso más en este sector, lanzando la versión para dispositivos embebidos en la Conferencia para desarrolladores de este tipo de sistemas que se celebra en Las Vegas. Esta decisión parece ir en consonancia con las predicciones que apuntan a una progresiva inclusión de microprocesadores en casi cualquier tipo de aparatos (frigoríficos, coches, etc).

DOS está estrechamente ligado al progresivo desarrollo de los microprocesadores de arquitectura x86 de Intel. Y es que es un sistema operativo que hay que conocer porque sigue estando en vigor gracias a su estabilidad ya que se desarrollaba en paralelo con la tecnología de Intel e IBM y porque es un sistema operativo muy simple, monousuario y monotarea. En otras palabras, DOS sólo ejecuta un programa cada vez

Java para sistemas embebidos. La tecnología Java es cada vez más utilizada en sistemas embebidos avanzados, debido a sus capacidades inherentes de soporte de red, optimización de dispositivos y procesamiento de datos. La mayoría de las características de la plataforma Java SE pueden ser ahora empleadas para el desarrollo embebido, gracias a la capacidad cada vez mayor del nuevo hardware disponible en el mercado.

SISTEMAS OPERATIVOS CE

Windows CE fue introducido en un conjunto de productos de PC de mano en el año 1996, pero posteriormente se transformó en un sistema operativo embebido altamente configurable. Entre sus ventajas se encuentran que incluye un subconjunto del API Win32 enfocado a los servicios que comúnmente se necesitan, así como una optimización en el consumo de energía. En este sistema cuando se habla de tamaño, se refiere a la cantidad de memoria y de almacenamiento necesario para albergar al sistema.

Mientras se desarrollaba Windows CE se le dio el nombre de Pegasus y junto a él se estaba diseñando un nuevo dispositivo que lo tendría más tarde como sistema operativo. A este dispositivo se le llamó

Handheld PC (HPC) o PC de bolsillo. CE, adicionando un conjunto completo de herramientas de desarrollo para la construcción de productos de diseño sofisticados, con aplicaciones poderosas para rodar en el hardware más actual.

Es un sistema operacional vigoroso, en tiempo real, para el desarrollo ágil de dispositivos de última generación conectados, inteligente y compacto. Éste incluye el software necesario para crear dispositivos basados en el Windows

Características

- Windows CE da un soporte bastante completo a las características más avanzadas que encontramos en computadores empujados, tanto de los procesadores como de otros elementos hardware.
- Es muy modular
- La API de Windows CE está pensada para facilitar la portabilidad con los sistemas operativos Windows para SSPG.

Plataforma

Windows CE soporta más de 160 modelos de procesadores, pertenecientes a 5 tipos de arquitecturas distintas: ARM, MIPS, PowerPC, SH y x86 (modo protegido). Son procesadores con una serie de características esenciales para este SO:

La cantidad de memoria necesaria para almacenar el sistema operativo depende de los módulos y funciones que necesite la plataforma. La versión mínima del SO, que incluye el kernel y sistema de ficheros (servicios

que veremos en la siguiente sección), ocupa unos 200 KB si se compila para un x86 en versión Release (400 KB en versión Debug). Una configuración con el máximo número de módulos puede llegar casi a los 30 MB en una compilación en modo debug para procesadores de la familia MIPS.

SISTEMAS OPERATIVOS TIEMPO REAL

Un proceso o tarea de tiempo real es el que se ejecuta en conexión con algún proceso, función o conjunto de sucesos externos al sistema informático y que debe cumplir uno o más plazos para interactuar en forma correcta y eficiente con el entorno exterior.

Características

- Pequeño tamaño (con una mínima funcionalidad asociada)
- Cambios de contexto rápidos
- Capacidad para responder rápidamente a interrupciones externas
- Multitarea con herramientas de comunicación entre procesos, como semáforos y señales
- Uso de archivos secuenciales para almacenar datos a alta velocidad
- Planificación expropiativa basada en prioridades
- Reducción de los intervalos en los que están deshabilitadas las interrupciones
- Alarmas especiales y temporizadores

ESTRUCTURA DE UN SISTEMA EMBEBIDO

Las principales características de un sistema embebido son el bajo costo y consumo de potencia. Dado que muchos sistemas embebidos son concebidos para ser producidos en miles o millones de unidades, el costo por unidad es un aspecto importante a tener en cuenta en la etapa de diseño. Generalmente, los sistemas embebidos emplean procesadores muy básicos, relativamente lentos y memorias pequeñas para minimizar los costos. En estos sistemas la velocidad no solo está dada por la velocidad del reloj del procesador, sino que el total la arquitectura se simplifica con el fin de reducir costos. Normalmente, un sistema embebido emplea periféricos controlados por interfaces seriales sincrónicas, las cuales son muchas veces más lentas que los periféricos empleados en un PC. Los sistemas embebidos deberán reaccionar a estímulos provenientes del ambiente, respondiendo con fuertes restricciones de tiempo en muchos casos, por lo tanto, un sistema se dice que trabaja en tiempo real si la información después de la adquisición y tratamiento es todavía vigente. Es decir, que en el caso de una información que llega de forma periódica, los tiempos de adquisición y tratamiento deben ser inferiores al período de actualización de dicha información. Un sistema embebido puede o no ser de tiempo de real dependiendo de los requerimientos específicos de la aplicación que se quiere implementar.

Los programas en estos sistemas se ejecutan minimizando los tiempos muertos y enfrentando fuertes limitaciones de hardware, ya que usualmente no tienen discos duros, ni teclados o monitores, una memoria flash reemplaza los discos y algunos botones y una pantalla LCD normalmente reemplazan los dispositivos de interfaz. El software que controla un dispositivo de

hardware, por ejemplo en una memoria ROM, Flash o un circuito integrado se conoce como Firmware. Típicamente la programación en estos dispositivos se realiza en lenguaje ensamblador o en lenguaje C, actualmente se han desarrollado algunas máquinas virtuales y otros compiladores que permiten el diseño de programas más complejos.

COMPONENTES DE UN SISTEMA EMBEBIDO

Un sistema embebido en principio estaría formando por un microprocesador y un software que se ejecute sobre este. Sin embargo este software necesitara sin duda un lugar donde poder guardarse para luego ser ejecutado por el procesador. Esto podría tomar la forma de memoria RAM o ROM, todo sistema embebido necesitara en alguna medida una cierta cantidad de memoria, la cual puede incluso encontrarse dentro del mismo chip del procesador. Además de esto normalmente un sistema embebido contara con una serie de salidas y entradas necesarias para comunicarse con el mundo exterior. Debido a que las tareas realizadas por sistemas embebidos son de relativa sencillez, los procesadores comúnmente usados cuentan con registros de 8 o 16 bits. En su memoria solo reside el programa destinado a gobernar una aplicación determinada. Sus líneas de entrada/salida soportan el conexionado de los sensores y actuadores del dispositivo a controlar y todos los recursos complementarios disponibles tienen como única finalidad atender a sus requerimientos. Estas son las únicas características que tienen en común los sistemas embebidos, todo lo demás

será totalmente diferente para cada sistema embebido en particular debido a la inmensa diversidad de aplicaciones disponibles.

PROCESO DE INSTALACIÓN

Los pasos necesarios para la instalación de los sistemas embebidos son:

1. Conocer la arquitectura hardware del dispositivo.
2. Configurar el kernel incluyendo los controladores para esos dispositivos y eliminar lo que no sirve.
3. Crear el script de configuración y ejecución de la aplicación embebida.
4. Configurar y compilar la busybox incluyendo la aplicación y el script.
5. Compilar el kernel
6. Prueba de ejecución

La arquitectura

Para generar el kernel correcto es necesario conocer lo más posible la arquitectura subyacente. Si disponemos del manual o datasheet del dispositivo podemos determinar los controladores necesarios. Si este no es nuestro caso, podemos arrancar con una distribución linux live (Ej. Ubuntu) y determinar que controladores son los necesarios analizando los módulos cargados y el árbol de ficheros generados bajo /sys.

Configuración del Kernel

El configurar correctamente el kernel es necesario para que tenga una huella pequeña pero con toda la funcionalidad requerida.

1. Seleccionar la arquitectura “más cercana/compatible” .
2. Deshabilitar “Enable loadable modules support”, para inscrutar los controladores dentro de la imagen y no depender de módulos externos.
3. Seleccionar el modo apropiativo. Si ese el caso se necesita poca latencia, luego se selecciona el modo Apropiativo “Preemption Model->Preemptible Kernel. Si por el contrario el sistema procesa transacciones a modo de servidor es conveniente usar “No force preemption”. Por defecto el kernel usa el término medio, Voluntary Preemption.
4. Reloj. Si la aplicación necesita temporización de alta precisión se debe de activar el temporizador de alta resolución.
5. Dispositivos de bloque. Activar unicamente los controladores SATA/PATA/IDE de el dispositivos.
6. USB. Es habitual habilitar el uso de USB 2.0 y la clase USB HID para poder usar teclados, ratones, etc. Se deshabilita todo lo demás.
7. Sistema de archivos. Seleccionamos el sistema de archivos que necesitemos (recomendable sistema de archivos con Jourling, ej. ext3 o reiserfs). En sistemas con poco disco suele usarse a menudo el JFFS2 como sistemas de archivos para memorias flash.
8. Video Linux. Normalmente se deshabilita.
9. sonido

Después de seleccionar los controladores necesarios se procede a generar una imagen del kernel “normal”. En este caso se debe construir un kernel que después de inicializarse no monte un sistema de archivos dentro de un disco duro (procedimiento normal), sino que monte un sistema de archivos temporal que va comprimido y embebido dentro de la propia imagen del kernel.

Para lograr esto se debe especificar un fichero especial que sirve de guía para generar este sistema de archivos.

Aplicación embebida

El objetivo de generar este kernel es poder correr la aplicación en un sistema embebido. Se debe procurar generar la aplicación teniendo muy en cuenta la arquitectura subyacente y aprovechando todo su potencial. Es recomendable si procede, generar un ejecutable estático e incluir herramientas de depuración remota o diagnósticos (ejecución, estado temperatura, carga de trabajo, logs, etc).

Una vez compilado lo agregamos a la lista de construcción de imagen la imagen con la correspondiente entrada “file”. El fichero init invocará la aplicación al final de la configuración.

Compilación del kernel.

Ya que se puede construir la imagen del kernel de forma habitual (“make”) o según los requisitos (compilación cruzada), se la podrá implementar en el sistema embebido.

GESTIÓN DE MEMORIA

Cualquiera de las funcionalidades de los sistemas embebidos están compuestas de tres aspectos: procesamiento, almacenamiento y comunicación. El procesamiento es la transformación de los datos, el almacenamiento es la retención de los datos para su posterior uso, y la comunicación es la transferencia de los datos. Cada uno de estos aspectos debe ser implementado. Se usan procesadores para el procesamiento, memoria para el almacenamiento, y buses para la comunicación

La mayoría de los sistemas operativos modernos convencionales usan memoria virtual paginada, donde la página es la unidad de protección y asignación de la memoria. El uso de procesos y protección de memoria en sistemas embebidos es muy importante; si se utiliza un único espacio de direcciones para todas las aplicaciones, una falla de software de una aplicación puede resultar en la corrupción de la memoria, ocasionando una falla de sistema. La desventaja, sin embargo, es que la protección de memoria requiere que el CPU soporte MMU (Memory Management Unit), lo cual resulta en un CPU más complejo.

A diferencia de los sistemas operativos convencionales, la mayoría de los sistemas operativos embebidos están enfocados a un CPU simple, que usualmente no tiene MMU. Además el resto del sistema tiene memoria limitada, poco o ningún espacio en disco; así que usualmente no usan memoria virtual.

Por ejemplo, los sistemas Linux embebido y QNX no soportan paginación, esto quiere decir que los datos, texto y pila comparten un espacio contiguo de memoria. Esto implica que no existe protección a nivel de

memoria, por ejemplo, la pila podría crecer hasta ocupar el espacio de texto o datos; o un proceso podría leer o escribir datos de otro proceso.

Por otro lado, Windows CE tiene un manejo de memoria más elaborado. Soporta parcialmente memoria virtual paginada (esto implica soporte de TLB (Translation Lookaside Buffer) en el CPU; y brinda protección de memoria a nivel de procesos e hilos.

Memoria real

La memoria real o principal es en donde son ejecutados los programas y procesos de una computadora y es el espacio real que existe en memoria para que se ejecuten los procesos. Por lo general esta memoria es de mayor costo que la memoria secundaria, pero el acceso a la información contenida en ella es de más rápido acceso. Solo la memoria cache es más rápida que la principal, pero su costo es a su vez mayor.

GESTIÓN DE ENTRADA Y SALIDA, DISPOSITIVOS QUE SOPORTA

Una de las principales funciones del sistema operativo es la gestión de los recursos de la computadora y, en concreto, de los dispositivos periféricos. El gestor de Entrada/Salida debe controlar el funcionamiento de todos los dispositivos de Entrada/Salida para alcanzar los siguientes objetivos:

- Facilitar el manejo de los dispositivos periféricos. Para ello ofrecer una interfaz sencilla, uniforme y fácil de utilizar entre los dispositivos, y gestionar los errores que se pueden producir en el acceso a los mismos.

- Ofrecer mecanismos de protección que impidan a los usuarios acceder sin control a los dispositivos periféricos.

Las rutinas que el sistema emplea para ejecutar operaciones de E/S están diseñadas para eliminar las diferencias entre los dispositivos y los tipos de acceso. No existe distinción entre acceso aleatorio y secuencial, ni hay un tamaño de registro lógico impuesto por el sistema.

El sistema mantiene una lista de áreas de almacenamiento temporal (buffers), asignadas a los dispositivos de bloques. El Kernel usa estos buffers con el objeto de reducir el tráfico de E/S. Cuando un programa solicita una transferencia, se busca primero en los buffers internos para ver si el bloque que se requiere ya se encuentra en la memoria principal (como resultado de una operación de lectura anterior).

SOPORTE A TIEMPO REAL

El soporte de red es importante en sistemas embebidos ya que les facilita comunicarse con el mundo exterior, así como su actualización. QNX contiene comunicación de red de bajo nivel en su microkernel; Windows CE posee varias pilas de comunicación a nivel de kernel (IP, PPP, IrDA, etc.).

Linux embebido por su parte posee un soporte de red heredado de los ambientes cliente/servidor soportados por Linux, ofreciendo pilas de red y protocolos de Internet.

Un sistema embebido complejo puede utilizar un sistema operativo como apoyo para la ejecución de sus programas, sobre todo cuando se requiere la ejecución simultánea de los mismos. Cuando se utiliza un sistema operativo lo más probable es que se tenga que tratar de un sistema operativo de tiempo real (RTOS).

Un sistema operativo de tiempo real (SOTR o RTOS -Real Time Operating System en inglés), es un sistema operativo que ha sido desarrollado para aplicaciones de tiempo real. Como tal, se le exige corrección en sus respuestas bajo ciertas restricciones de tiempo. Si no las respeta, se dirá que el sistema ha fallado. Para garantizar el comportamiento correcto en el tiempo requerido se necesita que el sistema sea predecible (determinista).

Características Generales

- No utiliza mucha memoria
- Cualquier evento en el soporte físico puede hacer que se ejecute una tarea
- Multi-arquitectura (puertos de código para otro tipo de UCP)
- Muchos tienen tiempos de respuesta predecibles para eventos electrónicos

En la actualidad hay un debate sobre qué es tiempo real. Muchos sistemas operativos de tiempo real tienen un programador y diseños de controladores que minimizan los periodos en los que las interrupciones están

deshabilitadas, un número llamado a veces duración de interrupción. Muchos incluyen también formas especiales de gestión de memoria que limitan la posibilidad de fragmentación de la memoria y aseguran un límite superior mínimo para los tiempos de asignación y retirada de la memoria.

ANÁLISIS COMPARATIVO CON LINUX Y WINDOWS 7.

Windows ce presenta un sistema dividió en 220modulos, cada uno dividido en varios componentes, los cuales son ROMables y comprimibles, ademas soporta cinco procesadores ARM/StrongARM, MIPS, PPC, SuperH, y x86 y un sobconjunto de la API Win 32, en cuanto a conectividad cuenta con el ActiveSync, que es la sincronización activa netre el PC de sobremesa y el dispositivo CE. Y también posse el remote API, un control remoto del dispositivo desde una aplicación en el PC.

En cuanto a Linux sabemos que es un sistema libre, cualquiera lo puede usar, modificar y descubrir, es además muy estable y es difícil que se quede colgado, los servidores que lo usan pueden funcionar durante meses sin parar, es extremadamente seguro, tiene varios sistemas de protección y no existe virus para Linux.

Ahora bien si hablamos de Windows 7 sabemos que pertenece a Microsoft, única compañía que lo puede modificar, las licencias son demasiado costosas, es poco estable, es común verse obligado a reiniciar el sistema, los servidores no admiten mas allá de un par de

semanas sin reiniciar, además de ser muy poco seguro y para el cual existen miles de virus que vulneran el sistema, solo lee y escribe sus propios sistemas de archivos y presenta incompatibilidades entre algunas de sus versiones.

La comparación de Microsoft Windows y el sistema operativo de Linux es un tema común de la discusión entre sus usuarios. Windows es el sistema operativo más importante bajo una licencia propietaria de software, mientras que Linux es el sistema operativo más importante bajo una licencia libre de software. Sin embargo, la mayoría de los sitios de distribuciones de Linux también disponen de componentes propietarios (tales como los controladores compilados de blob binario, que proveen los fabricantes de hardware para su instalación normal)

En noviembre de 2007, Linux era el sistema operativo del 85% de las supercomputadoras más poderosas (comparado con el 1,2% de Windows. En febrero de 2008, cinco de las diez empresas más confiables proveedoras de internet usaban Linux, mientras que sólo dos usaban Windows.

Los defensores de software libre discuten que la fuerza clave de Linux es que respeta lo que ellos consideran para ser las libertades esenciales de usuarios: la libertad de correrlo, Para estudiar y cambiarlo, y para redistribuir copias con o sin cambios.

OTROS ASPECTOS

WINDOWS EMBEDDED ¿UNA OPCIÓN COMO SISTEMA OPERATIVO DE ESCRITORIO?

Un sistema “Embedded” es un equipo con un propósito determinado, integrado en el sistema que controla. Necesita especificaciones particulares y realiza tareas predefinidas. Embedded significa encajado, embutido, algo metido dentro de otra cosa. Es un sistema operativo para plataformas de 32 bits adaptables y ampliables, que demanden conectividad y amplio soporte para el desarrollo de aplicaciones. Es la versión industrial del Windows XP Profesional. Basado en los mismos binarios que el Windows XP Profesional. Se eligen solo las funciones y servicios que se necesitan. Herramientas rápidas y potentes para la creación de imágenes propias para cada aplicación. Este sistema operativo es el que usan los ATM (cajeros automáticos), surtidores, puntos de venta, algunas consolas y maquinas de videojuegos, etc.

Para el desarrollo de Windows XP Embedded existen una serie de herramientas:

Target Analyzer(TA): Genera información sobre la composición hardware.

Component Designer(CD): Creación de componentes personalizados.

Target Designer(TD): Selección y configuración de los componentes para generar la imagen final S.O. Se usa para crear y generar el sistema

Windows Embedded es una versión de la plataforma Windows construida en base a módulos, lo que permite crear un sistema a medida. Si Windows XP o Vista

incluyen gran cantidad de programas como el reproductor de medios Windows Media Player o el navegador web Internet Explorer, Windows Embedded deja fuera todo esto para permitir utilizarlo como módulo, de forma que a aquellos clientes que no les interese el reproductor, pueden dejarlo fuera de su sistema.

El uso habitual de Windows Embedded es para la creación de sistemas empotrados; un ejemplo podría ser un navegador GPS, en el cual se quiere ofrecer un rango de funciones restringido pero a la vez que pueda actualizarse por software con cierta comodidad cuando así lo quiera el fabricante. Es por ello que este sistema se encuentra a la venta solamente para fabricantes de hardware, aunque en Internet disponemos de una demo que nos permite utilizar una instalación básica durante 120 días.

APLICACIONES DE LOS SISTEMAS OPERATIVOS EMBEBIDOS

Los lugares donde se pueden encontrar los sistemas empotrados son numerosos y de varias naturalezas. A continuación se exponen varios ejemplos para ilustrar las posibilidades de los mismos:

- Puntos de servicio o venta (POS, Point Of Service). Las cajas donde se paga la compra en un supermercado son cada vez más completas, integrando teclados numéricos, lectores de códigos de barras mediante láser, lectores de tarjetas bancarias de banda magnética o chip, pantalla alfanumérica de cristal líquido, etc. El sistema

empotrado en este caso requiere numerosos conectores de entrada y salida y unas características robustas para la operación continuada.

- Puntos de información al ciudadano. En oficinas de turismo, grandes almacenes, bibliotecas, etc. existen equipos con una pantalla táctil donde se puede pulsar sobre la misma y elegir la consulta a realizar, obteniendo una respuesta personalizada en un entorno gráfico amigable.

- Decodificadores y set-top boxes para la recepción de televisión. Cada vez existe un mayor número de operadores de televisión que aprovechando las tecnologías vía satélite y de red de cable ofrecen un servicio de televisión de pago diferenciado del convencional. En primer lugar envían la señal en formato digital MPEG-2 con lo que es necesario un procesamiento para decodificarla y mandarla al televisor. Además viaja cifrada para evitar que la reciban en claro usuarios sin contrato, lo que requiere descifrarla en casa del abonado. También ofrecen un servicio de televisión interactiva o web-TV que necesita de un software específico para mostrar páginas web y con ello un sistema basado en procesador con salida de señal de televisión.

- Sistemas radar de aviones. El procesamiento de la señal recibida o reflejada del sistema radar embarcado en un avión requiere alta potencia de cálculo además de ocupar poco espacio, pesar poco y soportar condiciones extremas de funcionamiento (temperatura, presión atmosférica, vibraciones, etc.).

- Equipos de medicina en hospitales y ambulancias UVI – móvil.
- Máquinas de revelado automático de fotos.
- Cajeros automáticos.
- Pasarelas (Gateways) Internet-LAN.

MARCO TEORICO

Como parte de la investigación a realizar sobre los sistemas operativos embebidos es necesario tener una base de conocimiento que responda a incógnitas singulares como lo son:

¿Qué son sistemas operativos embebidos?, ¿Que características presentan?, ¿Cómo se comportan? y otros interrogantes; cuyas respuestas las mostraremos a continuación.

Un sistema operativo embebido es aquel que ha sido creado para un sistema embebido, es decir, un sistema de computación limitado a un número fijo y escaso de tareas. Aquí no entran, evidentemente, los ordenadores personales ni los móviles, tabletas o dispositivos avanzados que conocemos.

Una peculiaridad de los sistemas embebidos es que son un todo con las aplicaciones que ejecutan, lo que quiere decir que en muchos casos no es posible instalar en ellos ningún tipo de software adicional.

Un uso muy común de los sistemas embebidos es en los sistemas de tiempo real, entendiéndose por sistemas en tiempo real a aquellos sistemas en los que el control del

tiempo es vital para el correcto funcionamiento. Los sistemas en tiempo real necesitan realizar ciertas operaciones o cálculos en un límite de tiempo. Donde ese límite de tiempo resulta crucial.

Algunas de las características de estos sistemas es que son dependientes. Dentro del concepto de dependencia surgen otros como confiabilidad, mantenimiento, seguridad y disposición. Estos aplican para sistemas embebidos que funcionan en fábricas o plantas industriales que corren durante todo el día. También, los sistemas embebidos tienen que ser eficientes, tanto en el consumo de energía como el código que ocupan para funcionar, esto se debe a la limitada capacidad de memoria de almacenamiento que poseen.

CONCLUSIONES

- El control del usuario es generalmente mucho mayor en un sistema operativo en tiempo real que en un sistema operativo ordinario. En sistema operativo típico que no sea en tiempo real, el usuario no tiene control sobre la función de planificación del sistema operativo. En un sistema en tiempo real resulta esencial permitir al usuario un control preciso sobre la prioridad de las tareas.
- A diferencia de los sistemas operativos convencionales, la mayoría de los sistemas operativos embebidos están enfocados a un CPU simple, que usualmente no tiene MMU. Además el resto del sistema tiene memoria limitada, poco o ningún espacio en disco; así que usualmente no usan memoria virtual.

OBSERVACIONES

- Los sistemas Linux embebido y QNX no soportan paginación, esto quiere decir que los datos, texto y pila comparten un espacio contiguo de memoria. Esto implica que no existe protección a nivel de memoria, por ejemplo, la pila podría crecer hasta ocupar el espacio de texto o datos; o un proceso podría leer o escribir datos de otro proceso.
- Para las personas que quieran entender de forma clara el funcionamiento de un sistema operativo embebido, primero deben estudiar como algunas implementaciones realizan el manejo de abstracciones básicas del sistema, de lo contrario les dificultara el entendimiento de estos sistemas operativos embebidos.

BIBLIOGRAFIA

- Asi son los sistemas operativos de cajeros, consultada el 18 de octubre de 2012 URL: <http://onsoftware.softonic.com/sistemas-embebidos-y-usos-cotidianos>
(<http://onsoftware.softonic.com/sistemas-embebidos-y-usos-cotidianos>)
- PDF Sistemas Embebidos y Sistemas Operativos Embebidos, universidad central de Venezuela, autor: David Perez. Consultado el 17 de octubre del 2012

- Sistema embebido, consultado el 18 de octubre de 2012 URL: http://es.wikipedia.org/wiki/Sistema_embebido
(http://es.wikipedia.org/wiki/Sistema_embebido)
- Sistemas de tiempo real y empotrado, consultado el 17 de octubre de 2012

URL: <http://es.scribd.com/doc/32547554/46/Windows-CE-caracteristicas>
(<http://es.scribd.com/doc/32547554/46/Windows-CE-caracteristicas>)



Esta entrada fue publicada en COMPILACION BIBLIOGRAFICA.

Añadir a marcadores el enlace permanente.

Deja un comentario

Blog de WordPress.com.